


10/11/01

PTO/SB/05 (08-00)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box 

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**UTILITY
PATENT APPLICATION
TRANSMITTAL**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. RSW919980041US2

First Inventor Dingsor et al.

Title Method and Apparatus Load Balancing Server Daemons within a Server

Express Mail Label No. EL750736463US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)

2. ☐ Applicant claims small entity status.
See 37 CFR 1.27.

3. ☒ Specification [Total Pages 36]
(preferred arrangement set forth below)

- Descriptive title of the invention
- Cross Reference to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to sequence listing, a table, or a computer program listing appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 9]

5. Oath or Declaration [Total Pages 2]

a. ☐ Newly executed (original or copy)

b. ☒ Copy from a prior application (37 CFR 1.63 (d))
(for continuation/divisional with Box 17 completed)

i. ☐ **DELETION OF INVENTOR(S)**
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

6. ☐ Application Data Sheet. See 37 CFR 1.76

7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)

8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)

a. ☐ Computer Readable Form (CRF)

b. Specification Sequence Listing on:

i. ☐ CD-ROM or CD-R (2 copies); or

ii. ☐ paper

c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. ☒ Assignment Papers (cover sheet & document(s))

10. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☒ Power of Attorney

11. ☐ English Translation Document (if applicable)

12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations

13. ☒ Preliminary Amendment

14. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)

15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)

16. ☐ Other:

17. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

☐ Continuation

☒ Divisional

☐ Continuation-in-part (CIP)

of prior application No.: 09, 161,905

Prior application information: Examiner Cardone, J. Group / Art Unit: 2152

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

18. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or ☒ Correspondence address below

Name Duke W. Yee

Address Carstens, Yee & Cahoon, L.L.P.
P.O. Box 802334

City Dallas

State TX

Zip Code 75380-2334


Country USA

Telephone (972) 367-2001

Fax (972) 367-2002

Name (Print/Type) Duke W. Yee

Registration No. (Attorney/Agent) 34,285

Signature 

Date 10/11/01

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2002

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT (\$ 740.00

Complete if Known

Application Number	Not Assigned
Filing Date	10/11/2001
First Named Inventor	Dingsor et al.
Examiner Name	Cardone, J.
Group Art Unit	2152
Attorney Docket No.	RSW919980041US2

METHOD OF PAYMENT

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number 09-0461
Deposit Account Name IBM Corporation

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Applicant claims small entity status See 37 CFR 1.27

2. ☐ Payment Enclosed:

☐ Check ☐ Credit card ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	740.00
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	

SUBTOTAL (1) (\$ 740.00

2. EXTRA CLAIM FEES

Total Claims	Extra Claims	Fee from below	Fee Paid
6	-20** = 0	18.00	0.00
2	-3** = 0	84.00	0.00
		280.00	0.00

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$ 0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for <i>ex parte</i> reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,280	241	640	Petition to revive - unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CFR 1.17(q)	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify)

**or number previously paid, if greater; For Reissues, see above

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$

SUBMITTED BY

Name (Print/Type)	Duke W. Yee	Registration No. (Attorney/Agent)	34,285	Complete (if applicable)
Signature	<i>D. Yee</i>	Telephone	(972) 367-2001	Date
				10/11/2001

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Dingsor et al.**

Serial No.: **Not Assigned**

Filed: **October 11, 2001**

For: **Method and Apparatus Load
Balancing Server Daemons within a
Server**

§
§
§
§
§
§

Group Art Unit: **2152**

Examiner: **Cardone, J.**

Attorney Docket No.: **RSW919980041US2**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as Express mail (EL750736463US) in an envelope addressed to: Assistant Commissioner for Patents, Box Patent Applications, Washington, D.C. 20231 on October 11, 2001.

By:

Krista Douthitt
Krista Douthitt

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

No fees are believed to be necessary at this time. However, in the event that any fees are required for the prosecution of this application, please charge any necessary fees to Deposit Account No. 09-0461. No extension of time is believed to be necessary. If, however, an extension of time is needed, the extension is requested.

Prior to examination of this application, please amend the above-identified application as follows:

IN THE SPECIFICATION:

On page one, before the BACKGROUND OF THE INVENTION, please insert the following paragraph:

This application is a divisional of application number 09/161,905, filed September 28, 1998, status pending.

IN THE CLAIMS:

Please cancel claims 1-15 and 21-37.

A clean version of the entire set of pending claims is as follows:

16. (Once Amended) A computer comprising:

- a plurality of processes, wherein the plurality of processes service a destination address and have process addresses;
- a packet routing layer, wherein the packet routing layer routes a packets to the plurality to the plurality of processes using a destination addresses within the packets;
- a dispatch layer between a TCP layer and an IP layer, wherein the dispatch layer has a plurality of modes of operation including:
 - a first mode of operation in which the dispatch layer receives a packet from a client, wherein the packet includes the destination address;
 - a second mode of operation, responsive to receiving the packet, in which the dispatch layer identifies a process within the plurality of processes to service the client, wherein the process is an identified process;
 - a third mode of operation in which the dispatch layer translates the destination address to a process address for the identified process within the plurality of processes;
 - and
 - a fourth mode of operation, responsive to the third mode of operation, in which the packet is sent to the packet routing layer.

17. (Unchanged) The computer of claim 16, wherein each packet includes a source address and wherein the dispatch layer further includes:

- a fifth mode of operation in which the dispatch layer receives a packet from the identified process for the client; and
- a sixth mode of operation, responsive to the fifth mode of operation, in which the dispatch layer translates the source address in the packet of the destination address.

18. (Unchanged) The computer of claim 16, wherein the second mode of operation includes:
a first submode of operation in which a determination is made as to whether a connection exists for the client;

a second submode of operation, responsive to a determination that a connection is absent for the client in the first submode of operation, in which a connection is created and a process within the plurality of processes is assigned to service the connection, wherein the process is the identified process; and

a third submode of operation, responsive to a determination that a connection exists for the client, in which a process assigned to the connection is identified and forms the identified process.

19. (Unchanged) The computer of claim 16, wherein the packet routing layer is a transmission control protocol layer.

20. (Unchanged) The computer of claim 16, wherein the plurality of processes is a plurality of server daemons.

38. (Once Amended) A computer program product for routing packets from a client to a selected process within a plurality of processes servicing a connection between the data processing system and the client comprising:

a computer readable medium;

first instructions for receiving a packet for the connection between the data processing system and the client, wherein the packet includes a destination address; and

second instructions for translating, in a dispatch layer between a TCP layer and an IP layer, the destination address to an intermediate destination address, which is an address for the selected process within the plurality of processes, wherein the instructions are embodied within the computer readable medium.

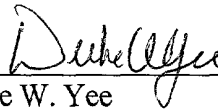
REMARKS

Claims 1-15 and 21-37 have been canceled. Claims 16-20 and 38 remain in the application. These claims are believed to be in condition for allowance. The amendments to claims 16-20 and 38 were made in the Response to Office Action as filed on June 26, 2001 in the parent case 09/161,905. No new matter has been added by these amendments.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Date: October 11, 2001

Respectfully submitted,



Duke W. Yee

Registration No. 34,285

CARSTENS YEE & CAHOON, LLP

P.O. Box 802334

Dallas, Texas 75380

(972) 367-2001

ATTORNEY FOR APPLICANT

Appendix of Claim Amendments

16. (Once Amended) A computer comprising:

a plurality of processes, wherein the plurality of processes service a destination address and have process addresses;

a packet routing layer, wherein the packet routing layer routes a packets to the plurality to the plurality of processes using a destination addresses within the packets;

a dispatch layer between a TCP layer and an IP layer, wherein the dispatch layer has a plurality of modes of operation including:

a first mode of operation in which the dispatch layer receives a packet from a client, wherein the packet includes the destination address;

a second mode of operation, responsive to receiving the packet, in which the dispatch layer identifies a process within the plurality of processes to service the client, wherein the process is an identified process;

a third mode of operation in which the dispatch layer translates the destination address to a process address for the identified process within the plurality of processes; and

a fourth mode of operation, responsive to the third mode of operation, in which the packet is sent to the packet routing layer.

38. (Once Amended) A computer program product for routing packets from a client to a selected process within a plurality of processes servicing a connection between the data processing system and the client comprising:

a computer readable medium;

first instructions for receiving a packet for the connection between the data processing system and the client, wherein the packet includes a destination address; and

second instructions for translating, in a dispatch layer between a TCP layer and an IP layer, the destination address to an intermediate destination address, which is an address for the selected process within the plurality of processes, wherein the instructions are embodied within the computer readable medium.

EXPRESS MAIL LABEL NO. EE023244743	DATE OF DEPOSIT: September 28, 1998
I hereby certify that this paper and fee are being deposited with the United States Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents, Washington, D.C. 20231.	
Gatherine M. Robbins	<i>Gatherine M. Robbins</i>
NAME OF PERSON MAILING PAPER AND FEE	SIGNATURE OF PERSON MAILING PAPER AND FEE

INVENTORS: Andrew D. Dingsor and Stephen M. Fontes

METHOD AND APPARATUS LOAD BALANCING SERVER DAEMONS WITHIN A SERVER

BACKGROUND OF THE INVENTION

Technical Field:

The present invention relates generally to an improved distributed data processing system, in particular to a method and apparatus for improving performance and availability of a server. Still more particularly, the present invention relates to method and apparatus for improving server performance and availability of a server in a distributed data processing system through binding server daemons within the server.

Description of Related Art:

Internet, also referred to as an "internetwork", in communications is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from the sending network to the protocols used by the receiving network (with packets if necessary). When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols. TCP/IP stands for Transmission Control Protocol/Internet Protocol. This protocol was developed by the Department of Defense for communications between computers. It is built into the UNIX system and has become the de facto standard for data transmission over networks, including the Internet.

The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty. Many federal, state, and local government agencies are also employing Internet sites for informational purposes, particularly agencies which must interact with virtually all segments of society such as the Internal Revenue Service and secretaries of state. Operating costs may be reduced by providing informational guides and/or searchable databases of public records online.

Currently, the most commonly employed mechanism of transferring data over the Internet is the World Wide Web

environment, also called simply "the web". Other Internet resources exist for transferring information, such as File Transfer Protocol (FTP) and Gopher, but have not achieved the popularity of the web. In the web environment, servers and clients effect data transaction using the Hypertext Transfer Protocol (HTTP), a known protocol for handling the transfer of various data files (e.g., text, still graphic images, audio, motion video, etc.). Information is formatted for presentation to a user by a standard page description language, the Hypertext Markup Language (HTML). In addition to basic presentation formatting, HTML allows developers to specify "links" to other web resources, including web sites, identified by a Uniform Resource Locator (URL). A URL is a special syntax identifier defining a communications path to specific information. Each logical block of information accessible to a client, called a "page" or a "web page", is identified by a URL. The URL provides a universal, consistent method for finding and accessing this information by the web "browser". A browser is a program capable of submitting a request for information identified by a URL at the client machine. Retrieval of information on the web is generally accomplished with an HTML-compatible browser, such as, for example, Netscape Communicator, which is available from Netscape Communications Corporation.

.A web site is typically located on a server, which in some cases may support multiple web sites. Many times, a web site can crawl when traffic on the web site is too heavy. As a result, popularity of a web site can be a detriment because the site cannot handle the amount

of traffic that the site is receiving. One mechanism used to increase performance of web site is to implement a server with more capacity and processing power or to employ multiple servers to handle the web site. With a larger server, a problem of single point failure is still present. If the server fails, the web site will be unavailable until the server can be repaired or replaced. Multiple servers are employed to solve that problem. With multiple servers, however, the performance of the web site may be increased, but individual servers may be under utilized. In addition, the contents for a web site are replicated on each server.

As a result, it is desirable to improve performance and availability of a server by load balancing among multiple server daemons running on one server with all server daemons responding to the same IP address and port number. Presently, however, additional capacity to support increased throughput on one server machine can only be achieved with multiple server daemons bound to different IP addresses or port numbers. This situation is due to an architectural basic limitation in TCP/IP, wherein TCP can deliver a received packet with a unique destination address and port number combination to only one server daemon.

Therefore, it would be advantageous to have an improved method and apparatus to increase availability and performance of a server without the different IP address and port number limitation.

SUMMARY OF THE INVENTION

5 The present invention provides a method and apparatus in a data processing system for binding a plurality of server daemons to a destination address and port. A request for a connection from a client is routed using a destination address. A server daemon within the plurality of server daemons is selected to form a selected server daemon. The request is routed to the selected server daemon by changing the destination address to a server address for the selected server daemon. When a response is returned, source address in the response is changed to the original destination address.

10

BRIEF DESCRIPTION OF THE DRAWINGS

5 The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

10 Figure 1 is a pictorial representation of a distributed data processing system in which the present invention may be implemented is depicted;

Figure 2 is a block diagram of a data processing system, which may be implemented as a server, in accordance to the present invention;

15 Figure 3 is a diagram of server system containing individual servers in which the processes of the present invention may be implemented in accordance with a preferred embodiment of the present invention;

20 Figure 4 is a diagram of data flow in a server in accordance with a preferred embodiment of the present invention;

Figure 5 is a diagram of an IP header in accordance with a preferred embodiment of the present invention;

Figure 6 is a diagram of a TCP header in accordance with a preferred embodiment of the present invention;

Figure 7 is a data structure in which information may be stored in responding to requests from clients in accordance with a preferred embodiment of the present invention;

Figure 8 is a flowchart of a process for handling inbound packets in accordance with a preferred embodiment of the present invention;

Figure 9 is a flowchart of a process for handling outbound packets in accordance with a preferred embodiment of the present invention; and

Figure 10 is a connection record table in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, and in particular with reference to Figure 1, a pictorial representation of a distributed data processing system in which the present invention may be implemented is depicted.

Distributed data processing system 100 is a network of computers in which the present invention may be implemented. Distributed data processing system 100 contains a network 102, which is the medium used to provide communications links between various devices and computers connected together within distributed data processing system 100. Network 102 may include permanent connections, such as wire or fiber optic cables, or temporary connections made through telephone connections.

In the depicted example, a server system 104 is connected to network 102 along with storage unit 106. Server system 104 may include one or more servers connected to each other in the depicted example. In addition, clients 108, 110, and 112 also are connected to a network 102. These clients 108, 110, and 112 may be, for example, personal computers or network computers. For purposes of this application, a network computer is any computer, coupled to a network, which receives a program or other application from another computer coupled to the network. In the depicted example, server system 104 provides data, such as, for example, boot files, operating system images, and applications to clients 108-112. Clients 108, 110, and 112 are clients to server system 104. Distributed data

processing system 100 may include additional servers, clients, and other devices not shown.

5 In the depicted example, distributed data processing system 100 is the Internet with network 102 representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, government, educational, and other computer systems, that route data and messages. Of course, distributed data processing system 100 also may be implemented as a number of different types of networks, such as for example, an intranet or a local area network.

10 Figure 1 is intended as an example, and not as an architectural limitation for the processes of the present invention.

15 Referring to Figure 2, a block diagram of a data processing system, which may be implemented as a server, is depicted in accordance to the present invention. In the instance that server system 104 is implemented as a single server, data processing system 200 may be used as the server. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors 202 and 204 connected to system bus 206. Alternatively, a single processor system may be employed. Also connected to system bus 206 is memory controller/cache 208, which provides an interface to local memory 209. I/O

bus bridge 210 is connected to system bus 206 and provides an interface to I/O bus 212. Memory controller/cache 208 and I/O bus bridge 210 may be integrated as depicted.

5 Peripheral component interconnect (PCI) bus bridge 214 is connected to I/O bus 212 and provides an interface to PCI local bus 216. A number of modems 218-220 may be connected to PCI local bus 216. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers 108-112 in Figure 1 may be provided through modem 218 and network adapter 220 connected to PCI local bus 216 through add-in boards.

10 Additional PCI bus bridges 222 and 224 provide interfaces for additional PCI buses 226 and 228, from which additional modems or network adapters may be supported. In this manner, server 200 allows connections to multiple network computers. A memory mapped graphics adapter 230 and hard disk 232 may also be connected to I/O bus 212 as depicted, either directly or indirectly.

15 Those of ordinary skill in the art will appreciate that the hardware depicted in Figure 2 may vary. For example, other peripheral devices, such as optical disk drive and the like also may be used in addition or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

20 The data processing system depicted in Figure 2 may be, for example, an IBM RISC/System 6000 system, a product of

International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system.

5 With reference now to Figure 3, a diagram of a server system is depicted in accordance with a preferred embodiment of the present invention. The processes of the present invention are used within a single data processing system, such as the computer illustrated in Figure 2. In
10 accordance with a preferred embodiment of the present invention, the processes of the present invention may be implemented in multiple servers as illustrated in Figure 3 in a tiered configuration with other load balancing mechanisms. Server system 104 from Figure 1 in the
15 depicted example is configured with a router 300, a load balancing data processing system 302 and servers 304-308. Corresponding reference numbers in different figures represent corresponding components unless specified otherwise. Router 300 receives packets destined for server system 104 from network 102. Load balancing data
20 processing system 302 routes packets received by router 300 to an appropriate server from servers 304-308. In the depicted example, load balancing data processing system 302 employs load balancing processes to maximize efficiency in processing requests from various clients. One or more of
25 the servers in servers 304-308 may implement the processes of the present invention. These servers may be implemented using a server such as data processing system 200 in Figure 2. The server system illustrated in Figure 3 is not
30 intended to imply architectural limitations to a server system implementation of the present invention.

5 The present invention provides a method, apparatus,
and instructions for binding multiple server daemons in a
server data processing system to the same IP address and
port number. In particular, when a packet is received by
the TCP/IP stack in a server, the TCP can pass the packet
up to only one daemon which is listening on that address
and port. The processes of the present invention changes
the destination internet protocol (IP) address in the
packet that is inbound or received by the server. An IP
10 address is a 32-bit (4-byte) binary number that uniquely
identifies a host (computer) connected to the Internet to
other Internet hosts, for the purposes of communication
through the transfer of packets. An IP address is
expressed in "dotted quad" format, consisting of the
15 decimal values of its four bytes, separated with periods;
for example, 127.0.0.1. The first one, two, or three
bytes of the IP address, assigned by InterNIC
Registration Services, identify the network the host is
connected to; the remaining bits identify the host
20 itself. The 32 bits of all 4 bytes together can signify
almost 232, or roughly 4 billion, hosts. This IP address
is changed back to its original value when the packet is
outbound or being sent out of the server. In this
manner, a number of different server daemons may be used
25 to handle packets destined for the same destination IP
address.

Although the processes described are for
implementation within a server, one or more of the
servers in Figure 3 may implement the processes and
30 instructions of the present invention in the
configuration shown in Figure 3 or in other

configurations in which multiple servers are placed into a tiered configuration with load balancing processes.

With reference now to **Figure 4**, a diagram of data flow in a server is depicted in accordance with a preferred embodiment of the present invention. This figure illustrates the flow of packets through a server, such as one used in server system 104. Although **Figure 4** provides an example using TCP, the present invention is not limited to TCP and may be applied to other transport layer protocols, such as, for example, User Datagram Protocol (UDP). Server 400 in **Figure 4** receives packets from clients and sends packets back to clients. A packet 402 is received at Internet Protocol (IP) layer 404. IP layer 404 incorporates the protocol within TCP/IP that governs the breakup of data messages into packets, the routing of the packets from sender to destination network and station, and the reassembly of the packets into the original data messages at the destination. IP corresponds to the network layer in the ISO/OSI model.

Packet 402 is processed by IP layer 404 and passed on to dispatch layer 406, which provides the routing mechanism used to route packets to different server daemons. In accordance with a preferred embodiment of the present invention, dispatch layer 406 is inserted between IP layer 404 and Transmission Control Protocol (TCP) layer 408. The mechanism incorporated within dispatch layer 406 allows for a number of server daemons, such as server daemons 410-414, to monitor or service the same IP address in fashion that is transparent to a

client. Dispatch layer 406 may change the destination IP address in packet 402 to route packet 402 to the appropriate server daemon if more than one server daemon is available to process requests made to the same destination IP address.

When packets are sent back from a server daemon for transmission to a client, dispatch layer 406 will change the destination IP address back to the original address. Dispatch layer 406 tracks the changes, if any, to the destination IP address so that the destination IP address may be changed back to the original address when a packet is to be returned to the client. This feature of changing and restoring the destination IP address is transparent to the client.

The packet sent from dispatch layer 406 to TCP layer 408, which includes the protocol within TCP/IP that governs the breakup of data messages into packets to be sent via IP, and the reassembly and verification of the complete messages from packets received by IP. TCP corresponds to the transport layer in the ISO/OSI model. TCP layer 408 sends the packet to a server daemon, such as server daemons 410-414, depending on the destination IP address in the packet.

When a packet, such as packet 416, is sent from one of the server daemons for transport to a client, TCP layer 408 will receive the packet and process it according to TCP protocols. Packet 416 is then sent to dispatch layer 406, which will determine if the

destination IP address for the incoming packet to the server daemon was changed. If the destination IP address was changed, the destination IP address in packet 416 is changed back to the original address. Packet 416 will then be passed to IP layer 404 for processing and then sent to the client.

Although the depicted example, dispatch layer 406 is located between TCP layer 408 and IP layer 404, dispatch layer 406 may be located in other places below TCP layer 408. For example, dispatch layer 406 could be located below IP layer 404. In addition, the processes in dispatch layer 406 could be implemented within IP layer 404 itself.

Turning to Figure 5, a diagram of an IP header is illustrated in accordance with a preferred embodiment of the present invention. This configuration 540 shows organization of an IP header and IP data area. Configuration 540 is shown as a sequence of 32 bit words. The first six words in the sequence are the IP header 544 and the remaining words are in IP data area 546. The numbers 542 across the top of the configuration 540 show the starting bit location of the various fields in the words of the IP message. The IP fields of particular interest are the protocol field 547, source IP address field 548, and destination address field 549. Each data processing system using IP is assigned a globally unique IP address. The protocol field 547 gives information about the protocol used in the next highest layer of protocol. For instance, the protocol field 547 specifies

if the next highest level will use UDP, TCP, or another protocol. The source IP address field 548 specifies the address of the computer which originated the message. The destination address field 549 specifies the address of the computer which is to receive the message. Other fields in the IP header, like total length and fragment offset, are used to breakup network datagrams into packets at the source computer and reassemble them at the destination computer. The header checksum is a checksum over the fields of the header, computed and set at the source and recomputed for verification at the destination.

Next in Figure 6, a diagram of a TCP header is illustrated in accordance with a preferred embodiment of the present invention. The TCP format is shown as a sequence of 32 bits words, the first six words being the TCP header 666 and the remaining words being the TCP data 667. The numbers 662 across the top of the TCP format 660 represent the starting bit locations of the fields in the TCP header 666. The TCP header 666, being a port type protocol, has port information in its header including a source port 663 and a destination port 664.

A TCP connection from a port on the source machine to a port on a destination machine is defined by four values: source address 648, source port 663, destination address 640, and the destination port 664 of the remote port of that machine. When the TCP protocol layer receives a TCP datagram, it uses all four values to determine which connection the data is for. Thus, on any one machine, TCP ensures that the set of active

connections is unique. TCP ports are not required to be unique. The same TCP port may be used in multiple connections, as long as those connections are unique.

5 The dispatcher is configured to store information describing which server daemons are available to be used when responding to requests from clients. One possible configuration implementation is shown in Figure 7 as a "tree view". Turning now to Figure 7, a data structure in which information may be stored in responding to
10 requests from clients is depicted in accordance with a preferred embodiment of the present invention. Data structure 700 relates the cluster address and port number, as targeted by the client, with the set of server daemon addresses within the server machine. Practical
15 configurations can range from a single cluster, single port, and two servers, to multiple clusters, multiple ports, and multiple servers. In addition, ports can be configured to support TCP only, UDP only, other protocols, or all protocols.

20 With reference now to Figure 8, a flowchart of a process for handling inbound packets is depicted in accordance with a preferred embodiment of the present invention. In the depicted example, this process is employed by dispatch layer 406 in Figure 4 to process a
25 packet received by a server having multiple daemons designated to handle the same destination IP address. The process begins by receiving a packet from the IP layer (step 800). A determination is made as to whether the destination IP address of the packet matches a ND cluster
30 address and whether the destination TCP port matches a ND

port. Step 802 is performed by consulting a ND configuration. In the depicted examples, a "cluster address" is an IP address known to clients as an address of the target. The cluster address is used by the client to access or send requests to the target. Referring to the dispatcher data structure in Figure 7, the destination IP address in the received packet is first considered. If the destination address matches a configured dispatcher cluster address, then the port numbers are considered. If the destination port number in the packet matches one of the port numbers configured under that cluster address, then the packet will be handled by the dispatcher.

Then, a determination is made as to whether the packet is part of an existing TCP connection (step 804). The determination in step 804 is made by consulting a connection record table maintained by dispatch layer 406, which is described in more detail below in Figure 10. If the packet is not part of an existing connection, a daemon server is selected (step 806). This selection may be made in a number of ways. The selections may be, for example, performed in a round robin mechanism or using any other known load balancing mechanisms. These load balancing mechanisms can range from simple round robin techniques, to sophisticated weighted round robin algorithms with active management processes. In the simple round robin case, each new client request is forwarded to the next server daemon in the configuration, regardless of the number of connections previously sent to that server, and regardless of whether that server is still responding properly. More sophisticated select

5 servers based upon information provided by their active
management processes. These processes monitor the number
of new connections sent to each server, the number of
connections presently active on each server, and the
health of each server (via other active processes which
periodically assess the response time or another
parameter of each server daemon). This information is
then combined and used by the load balancing algorithm to
select a server daemon within the server to service the
10 connection.

15 Using the selected server, a new record is then
added to the connection record table (step 808). Next,
the destination IP address of the packet is translated
from the cluster address (the current value of the
destination IP address in the packet) to the address of
the server daemon selected to process the connection
(step 810). A new IP checksum is then calculated for the
packet taking into account the destination IP address of
the server daemon (step 812). The packet is then
20 forwarded to TCP layer 408 (step 814).

25 With reference again to step 804, if the packet is
part of an existing connection, the address of the
previously selected server is selected from the table
(step 816) with the process then proceeding to step 810
as previously described. Referring back to step 802, if
the destination IP address and the destination TCP port
of the packet both do not match a ND cluster address and
a ND port, the packet is forwarded to TCP layer 408 in
step 814. This occurs when a packet is not destined for

a connection that has multiple server daemons assigned to handle the connection.

Turning next to Figure 9, a flowchart of a process for handling outbound packets is depicted in accordance with a preferred embodiment of the present invention. In the depicted example, this process is employed by dispatch layer 406 in Figure 4 to process a packet that is to be sent to a client from a server having multiple daemons designated to handle the same destination IP address.

The process begins by receiving a packet from the TCP layer (step 900). A determination is made as to whether the destination IP address of the packet matches one of the server addresses and whether the source TCP port matches a ND port. Step 902 is performed by consulting a ND configuration. If the source IP address matches the ND server address and the source TCP port matches a ND port, a determination is made as to whether the packet is part of an existing TCP connection (step 904). The determination in step 904 is made by consulting the connection record table, which is described in more detail below in Figure 10. If the packet is part of an existing connection, the cluster address is obtained from the table (step 906). In the depicted example, the cluster address is obtained from the connection record table. Then, the source IP address of the packet is translated from the server daemon address to the cluster address (step 908). A new IP checksum is then calculated for the packet taking into account the destination IP address of the server daemon

(step 910). The packet is then forwarded to IP layer 404 (step 912). IP layer 404 processes the packet for transmission to the client.

5 With reference again to step 904, if the packet is not part of an existing connection, the process then proceeds to step 912 as described above. Referring back to step 902, if the source IP address and the source TCP port of the packet both do not match a ND sever and a ND port, the packet is forwarded to IP layer 404 in step 10 912. This occurs when a packet is not destined for a connection that has multiple server daemons assigned to handle the connection. Some addresses may not be for server daemons or are not for connections that are assigned more than one server daemon.

15 Turning now to Figure 10, a connection record table is illustrated in accordance with a preferred embodiment of the present invent ion. Connection record table 1000 includes a number of records 1002, also referred to as entries. Each record 1002 includes form the received packet a source IP address 1004, a destination IP address 20 1006, a source TCP port 1008, and a destination TCP port 1010. Each record 1002 also includes the address of the server 1012 selected by the load balancing mechanism. Each record 1002 provides a mapping between the packets which 25 flow to and from the client and the packets which flow to and from the server daemons.

In connection with steps 804 and 808 in Figure 8, records are added to the connection record table 1000 each

time a new connection is established. The two IP addresses and two port numbers from the received packet are copied into in the new entry. Following the load balancing's selection of a server daemon, that server's address is then added to the connection table entry. Then, per step 810, the destination IP address field of the received packet is changed to that of the selected server. When a connection is terminated, the connection record is removed from the table. Garbage collection mechanisms are also used to purge from the table any stale connections which have not terminated gracefully. Connection record table 1000 is employed to track connections and server daemons used to service the connections.

Through the use of connection record table 1000, dispatch layer 406 can track which daemon is servicing a particular connection and make the appropriate destination IP address translations to route incoming packets to the appropriate server daemon. With outgoing packets, connection record table 1000 is used by dispatch layer 406 to restore the source IP address to the one that was used by the client. The restoration of the source IP address in outgoing packets provides seamless handling of packets in "binding" or assigning multiple daemons to the same IP address and port number. Thus, a number of daemons can listen on the same address and port to provide increased capacity within a single server.

Further, dispatch layer 406 can be used to support multiple IP destination addresses. Multiple cluster addresses with different groups of multiple server daemons may be handled using the processes of the present

invention. Thus, the present invention provides for scalable capabilities within a server.

5 It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in a form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog communications links.

10 The description of the present invention has been presented for purposes of illustration and description, but is not limited to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, the present invention is not limited to the traditional web server, using HTTP on port 80, but may be applied to support multiple protocols and/or port numbers. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1 1. A method in a data processing system for binding a
2 plurality of processes to a destination address, the
3 method comprising the data processing system implemented
4 steps of:
5 receiving a request for a connection, wherein the
6 request includes a destination address;
7 selecting a process from the plurality of processes
8 to form a selected process;
9 changing the destination address in the request to
10 an address for the selected process; and
11 sending the request to the selected process using
12 the address for the selected process.

1 2. The method of claim 1 further comprising:
2 detecting a response from the selected process,
3 wherein the response includes a source address; and
4 changing the source address to the destination
5 address.

1 3. The method of claim 1, wherein the plurality of
2 processes is a plurality of server daemons.

1 4. The method of claim 1, wherein further comprising:
2 creating a record in a table, wherein the record
3 provides mapping between packets flowing to and from a
4 client and between packets flowing to and from the
5 selected process.

1 5. The method of claim 4, wherein each record includes
2 a source IP address, a destination IP address, a source
3 TCP port, a destination TCP port, and an address of the
4 selected process.

1 6. The method of claim 1, wherein the step of selecting
2 a process is performed using a load balancing process.

1 7. A method in a data processing system for routing
2 packets from a client to a selected process within a
3 plurality of processes servicing a connection between the
4 data processing system and the client, the method
5 comprising the data processing system implemented steps
6 of:
7 receiving a packet for the connection between the
8 data processing system and the client, wherein the packet
9 includes a destination address; and
10 translating the destination address to an
11 intermediate destination address, which is an address for
12 the selected process within the plurality of processes.

1 8. The method of claim 7, wherein the translating step
2 is performed using a table that provides mapping between
3 packets which flow to and from the client and between
4 packets that flow to and from the plurality of processes.

1 9. The method of claim 7, wherein the plurality of
2 processes comprise a plurality of server daemons.

1 10. The method of claim 7, wherein the table includes a
2 record for each connection.

1 11. The method of claim 10 further comprising:
2 detecting a termination of a connection; and
3 removing a record from the table in response to a
4 termination of a connection associated with the record.

1 12. A method in a computer for routing packets between a
2 plurality of server daemons on the computer and a client,
3 the method comprising the computer implemented steps of:
4 receiving a packet from a client, wherein the packet
5 includes a destination address;
6 determining whether a connection exists for the
7 client;
8 responsive to a determination that a connection is
9 absent for the client, selecting a server daemon from the
10 plurality of server daemons to service the client,
11 wherein the plurality of server daemons service the
12 destination address in the packet and wherein the
13 plurality of server daemons have assigned destination
14 addresses; and
15 altering the destination address in the packet to an
16 assigned destination address for the server daemon.

1 13. The method of claim 12 further comprising:
2 sending the packet to the server daemon using the
3 assigned destination address.

1 14. The method of claim 13 further comprising:
2 receiving a return packet for the client from the
3 server daemon, wherein the return packet includes a
4 source address; and
5 altering the source address in the return packet to
6 that of the destination address.

1 15. The method of claim 12 further comprising:
2 responsive to selecting the server daemon, creating
3 a record in a table in which the record includes an
4 address of the server daemon and the destination address.

1 16. A system in a data processing environment
2 comprising:
3 a plurality of processes, wherein the plurality of
4 processes service a destination address and have process
5 addresses;
6 a packet routing layer, wherein the packet routing
7 layer routes a packets to the plurality of processes
8 using a destination addresses within the packets;
9 a dispatch layer, wherein the dispatch layer has a
10 plurality of modes of operation including:
11 a first mode of operation in which the dispatch
12 layer receives a packet from a client, wherein the packet
13 includes the destination address;
14 a second mode of operation, responsive to
15 receiving the packet, in which the dispatch layer
16 identifies a process within the plurality of processes to

17 service the client, wherein the process is an identified
18 process;
19 a third mode of operation in which the dispatch
20 layer translates the destination address to a
21 process address for the identified process within
22 the plurality of processes; and
23 a fourth mode of operation, responsive to the
24 third mode of operation, in which the packet is sent
25 to the packet routing layer.

1 17. The system of claim 16, wherein each packet includes
2 a source address and wherein the dispatch layer further
3 includes:
4 a fifth mode of operation in which the dispatch
5 layer receives a packet from the identified process for
6 the client; and
7 a sixth mode of operation , responsive to the
8 fifth mode of operation, in which the dispatch layer
9 translates the source address in the packet of the
10 destination address.

1 18. The system of claim 16, wherein the second mode of
2 operation includes:
3 a first submode of operation in which a
4 determination is made as to whether a connection exists
5 for the client;
6 a second submode of operation, responsive to a
7 determination that a connection is absent for the client
8 in the first submode of operation, in which a connection
9 is created and a process within the plurality of

10 processes is assigned to service the connection, wherein
11 the process is the identified process; and
12 a third submode of operation, responsive to a
13 determination that a connection exists for the client, in
14 which a process assigned to the connection is identified
15 and forms the identified process.

1 19. The system of claim 16, wherein the packet routing
2 layer is a transmission control protocol layer.

1 20. The system of claim 16, wherein the plurality of
2 processes is a plurality of server daemons.

1 21. A data processing system for binding a plurality of
2 processes to a destination address, the data processing
3 system comprising:
4 receiving means for receiving a request for a
5 connection, wherein the request includes a destination
6 address;
7 selection means for selecting a process from the
8 plurality of processes to form a selected process;
9 changing means for changing the destination address
10 in the request to an address for the selected process;
11 and
12 sending means for sending the request to the
13 selected process using the address for the selected
14 process.

1 22. The data processing system of claim 21, wherein the
2 changing means is a first changing means and further
3 comprising:

4 detection means for detecting a response from the
5 selected process, wherein the response includes a source
6 address; and

7 second changing means for changing the source
8 address to the destination address.

1 23. The data processing system of claim 21, wherein the
2 plurality of processes is a plurality of server daemons.

1 24. The data processing system of claim 21, wherein
2 further comprising:

3 creation means for creating a record in a table,
4 wherein the record provides mapping between packets
5 flowing to and from a client and between packets flowing
6 to and from the selected process.

1 25. The data processing system of claim 24, wherein each
2 record includes a source IP address, a destination IP
3 address, a source TCP port, a destination TCP port, and
4 an address of the selected process.

1 26. The data processing system of claim 21, wherein the
2 selection means is performed using a load balancing
3 process.

1 27. A data processing system for routing packets from a
2 client to a selected process within a plurality of
3 processes servicing a connection between the data
4 processing system and the client, the data processing
5 system comprising:

6 receiving means for receiving a packet for the
7 connection between the data processing system and the
8 client, wherein the packet includes a destination
9 address; and

10 translating means for translating the destination
11 address to an intermediate destination address, which is
12 an address for the selected process within the plurality
13 of processes.

1 28. The data processing system of claim 27, wherein the
2 translating means performed using a table that provides
3 mapping between packets which flow to and from the client
4 and between packets that flow to and from the plurality
5 of processes.

1 29. The data processing system of claim 27, wherein the
2 plurality of processes comprise a plurality of server
3 daemons.

1 30. The data processing system of claim 27, wherein the
2 table includes a record for each connection.

1 31. The data processing system of claim 30 further
2 comprising:

3 detection means for detecting a termination of a
4 connection; and

5 removing means for removing a record from the table
6 in response to a termination of a connection associated
7 with the record.

1 32. Computer readable code for routing packets between a
2 plurality of server daemons on a computer and a client,
3 said computer readable code comprising:

4 first subprocesses for receiving a packet from a
5 client, wherein the packet includes a destination
6 address;

7 second subprocesses for determining whether a
8 connection exists for the client;

9 third subprocesses, responsive to a determination
10 that a connection is absent for the client, for selecting
11 a server daemon from the plurality of server daemons to
12 service the client, wherein the plurality of server
13 daemons to service the destination address in the packet
14 and wherein the plurality of server daemons have assigned
15 destination addresses; and

16 fourth subprocesses for altering the destination
17 address in the packet to an assigned destination address
18 for the server daemon.

1 33. The computer readable code of claim 32 further
2 comprising:

3 fifth subprocesses for sending the packet to the
4 server daemon using the assigned destination address.

1 34. The computer readable code of claim 32 further
2 comprising:

3 sixth subprocesses for receiving a return packet for
4 the client from the server daemon, wherein the return
5 packet includes a source address; and

6 seventh subprocesses for altering the source address
7 in the return packet to that of the destination address.

1 35. The computer readable code of claim 32 further
2 comprising:

3 eighth subprocesses, responsive to selecting the
4 server daemon, for creating a record in a table in which
5 the record includes an address of the server daemon and
6 the destination address.

1 36. A computer program product for binding a plurality
2 of processes to a destination address, the computer
3 program product comprising:

4 a computer readable medium;

5 first instructions for receiving a request for a
6 connection, wherein the request includes a destination
7 address;

8 second instructions for selecting a process from the
9 plurality of processes to form a selected process;

10 third instructions for changing the destination
11 address in the request to an address for the selected
12 process; and
13 fourth instructions for sending the request to the
14 selected process using the address for the selected
15 process,
16 wherein the instructions are embodied within the
17 computer readable medium.

1 37. The computer program product of claim 36 further
2 comprising:
3 fifth instructions for detecting a response from the
4 selected process, wherein the response includes a source
5 address; and
6 sixth instructions for changing the source address
7 to the destination address.

1 38. A computer program product for routing packets from
2 a client to a selected process within a plurality of
3 processes servicing a connection between the data
4 processing system and the client comprising:
5 a computer readable medium;
6 first instructions for receiving a packet for the
7 connection between the data processing system and the
8 client, wherein the packet includes a destination
9 address; and
10 second instructions for translating the destination
11 address to an intermediate destination address, which is
12 an address for the selected process within the plurality
13 of processes,

14 wherein the instructions are embodied within the
15 computer readable medium

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

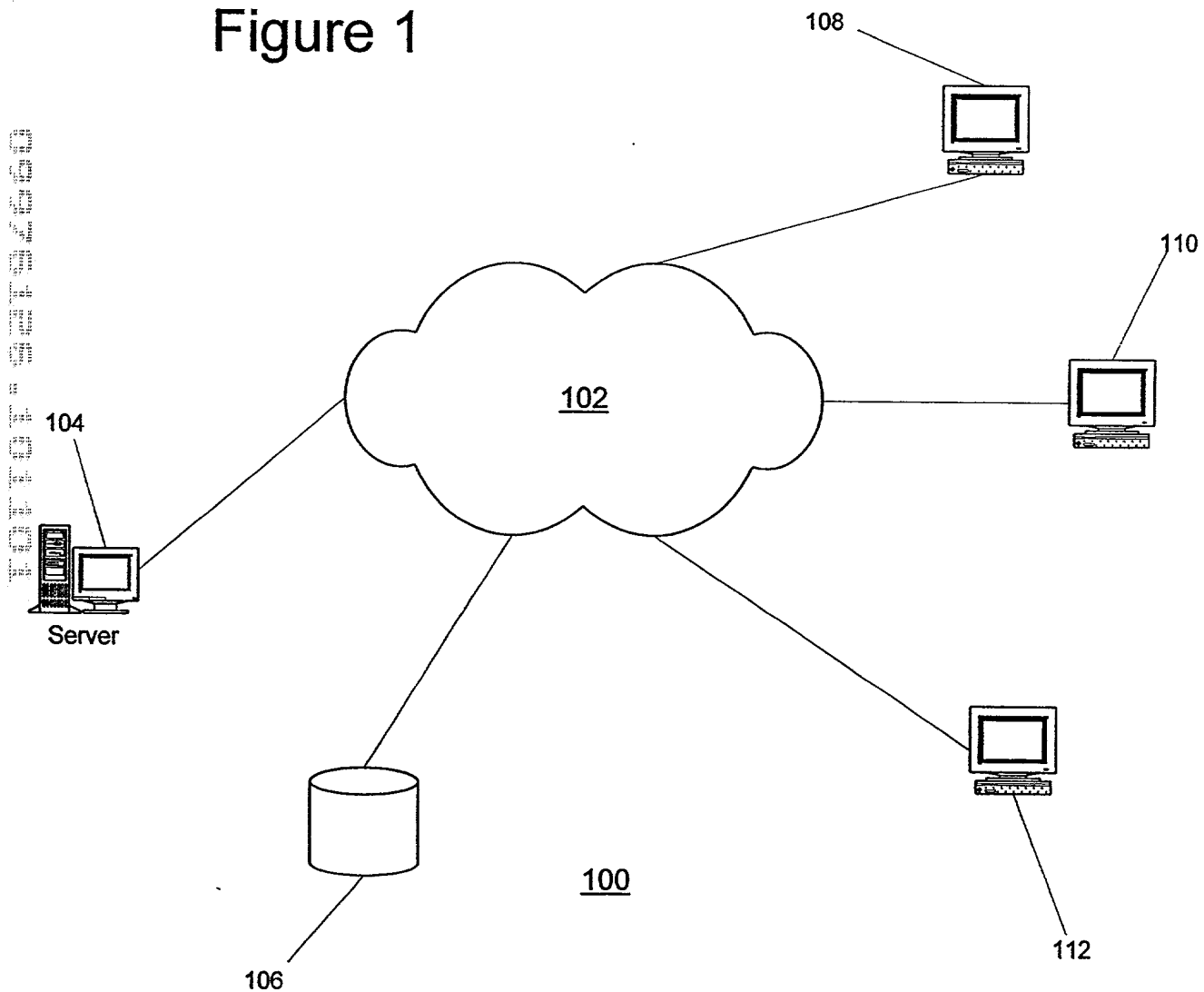
ABSTRACT

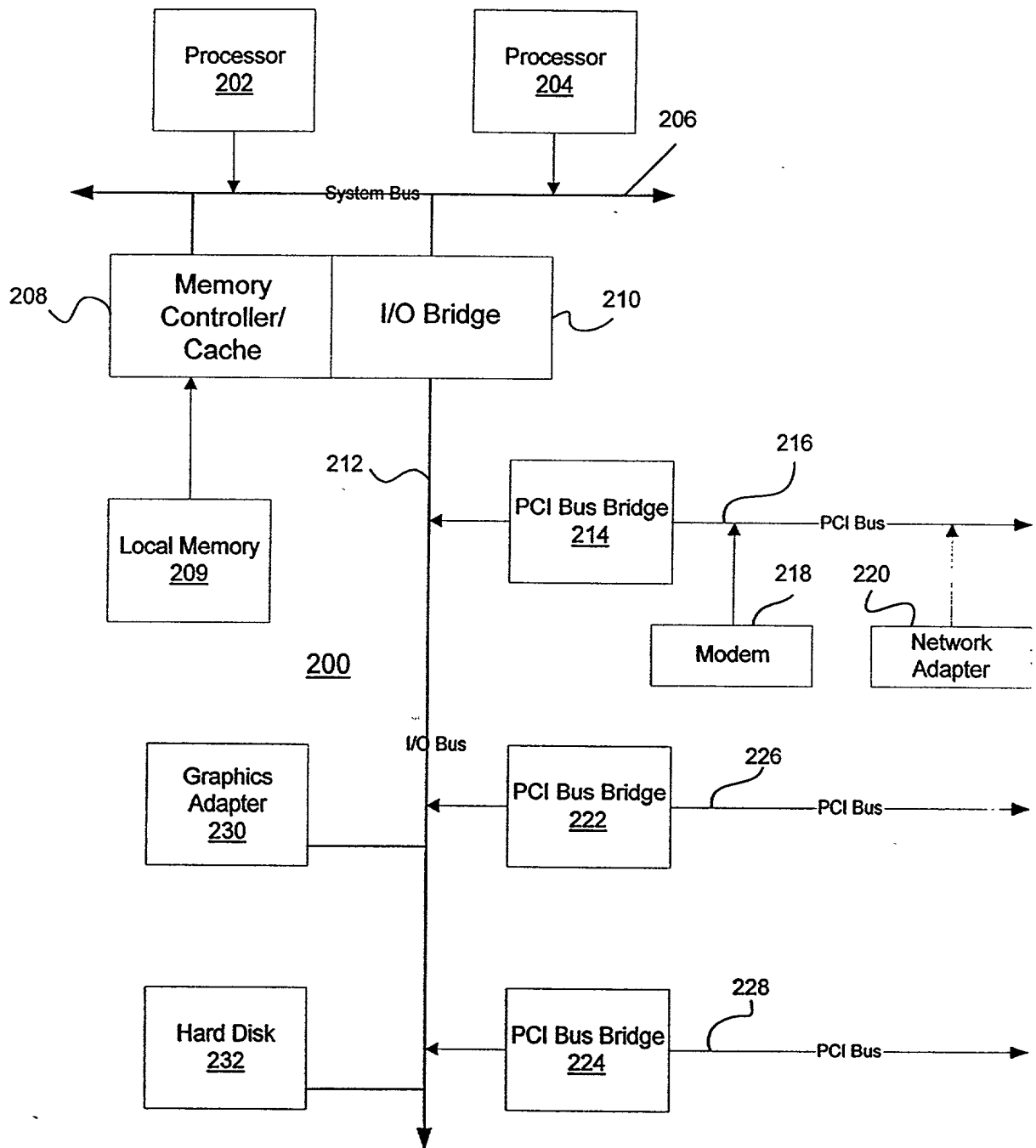
METHOD AND APPARATUS LOAD BALANCING SERVER DAEMONS WITHIN A SERVER

A method and apparatus in a data processing system
5 for binding a plurality of server daemons to a
destination address and port. A request for a connection
from a client is routed using a destination address. A
server daemon within the plurality of server daemons is
selected to form a selected server daemon. The request
10 is routed to the selected server daemon by changing the
destination address to a server address for the selected
server daemon. When a response is returned, source
address in the response is changed to the original
destination address.

Network

Figure 1





Sever

Figure 2

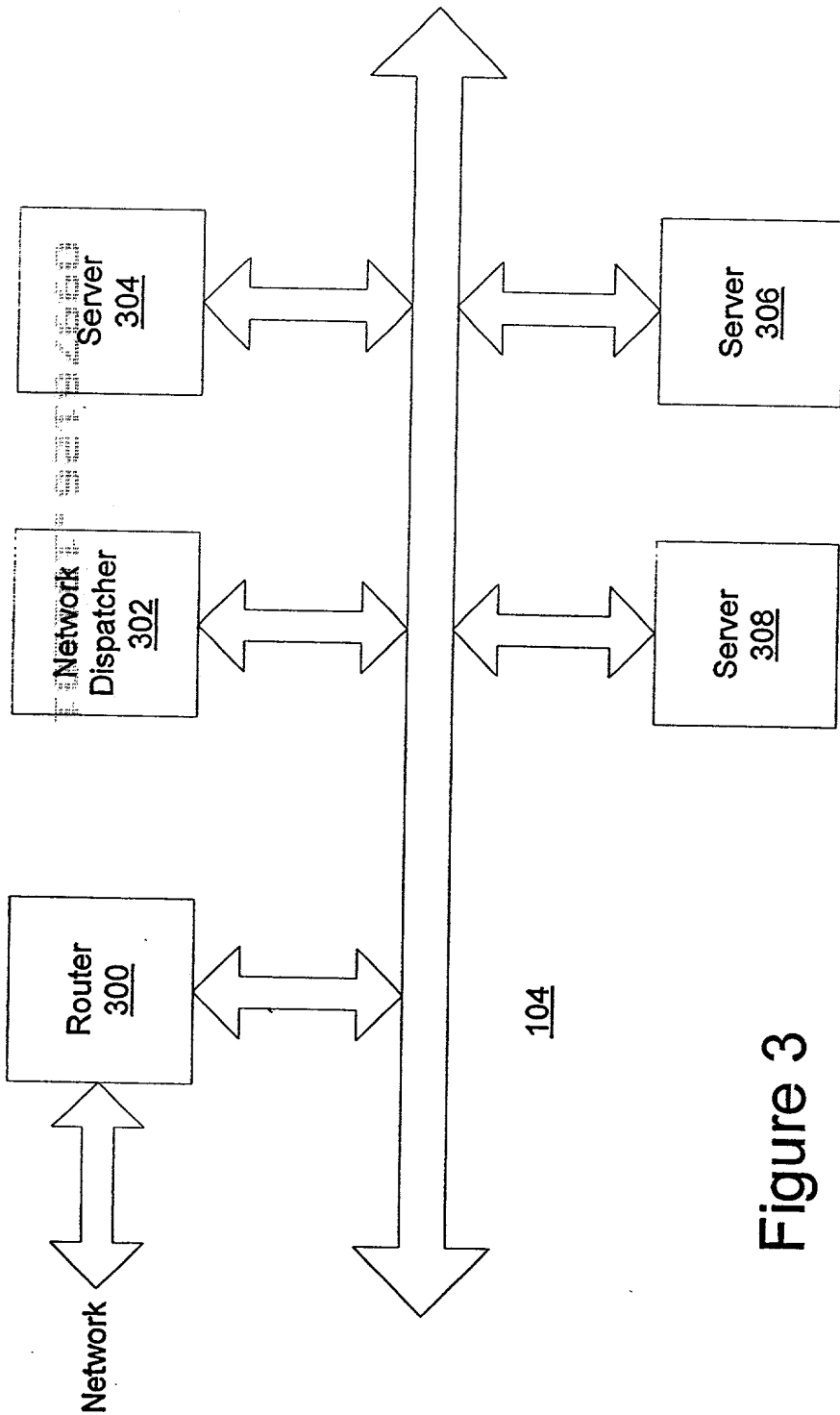


Figure 3

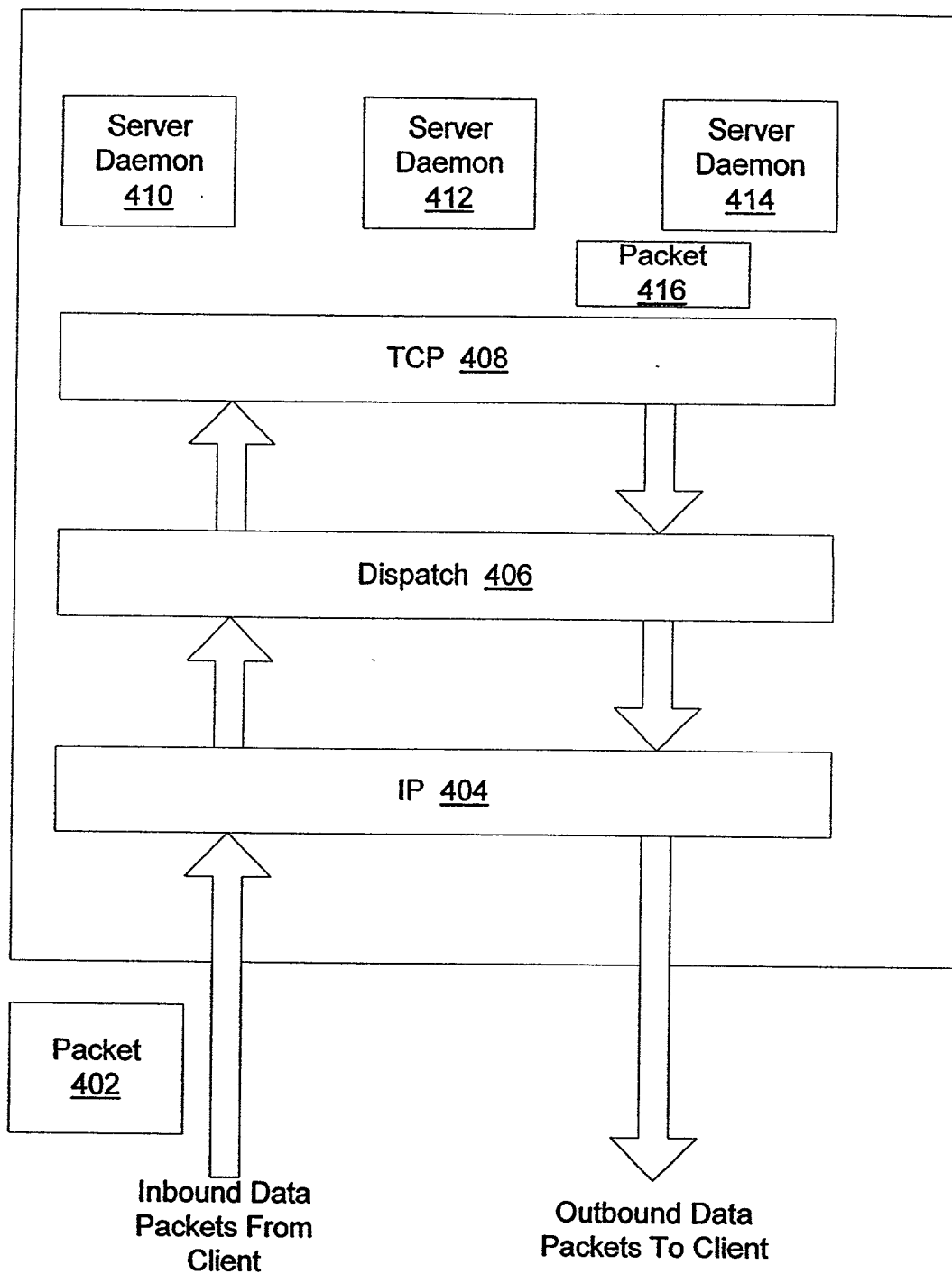


Figure 4

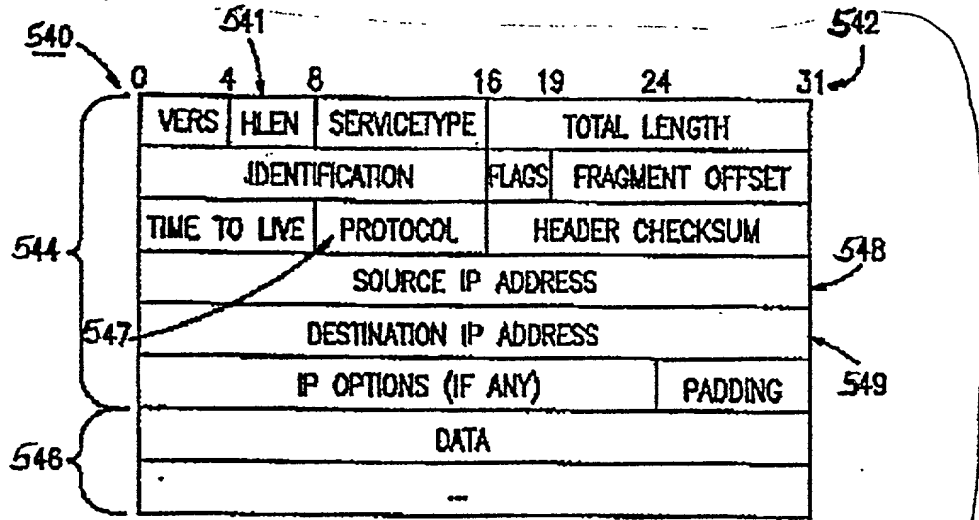


FIG. 5

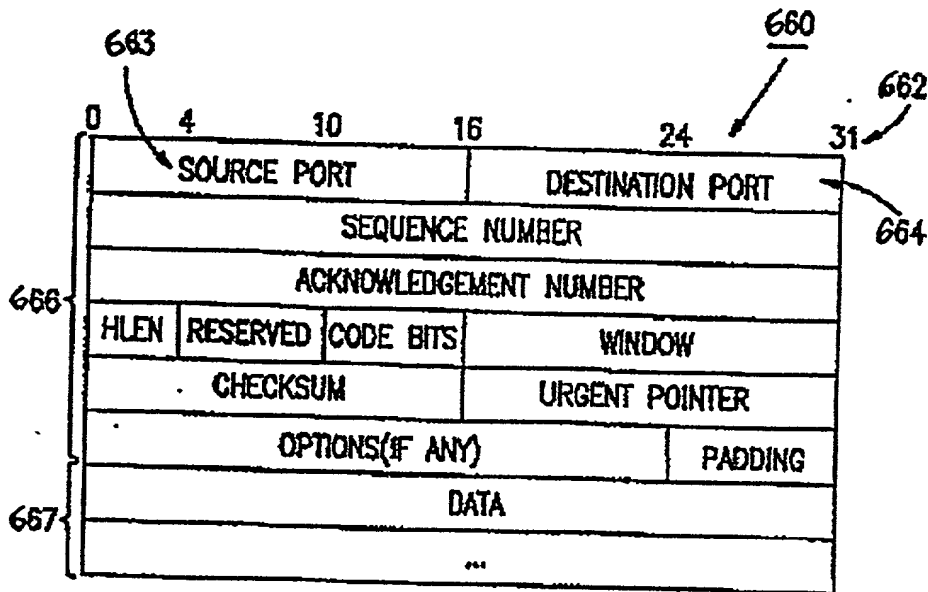


FIG. 6

700

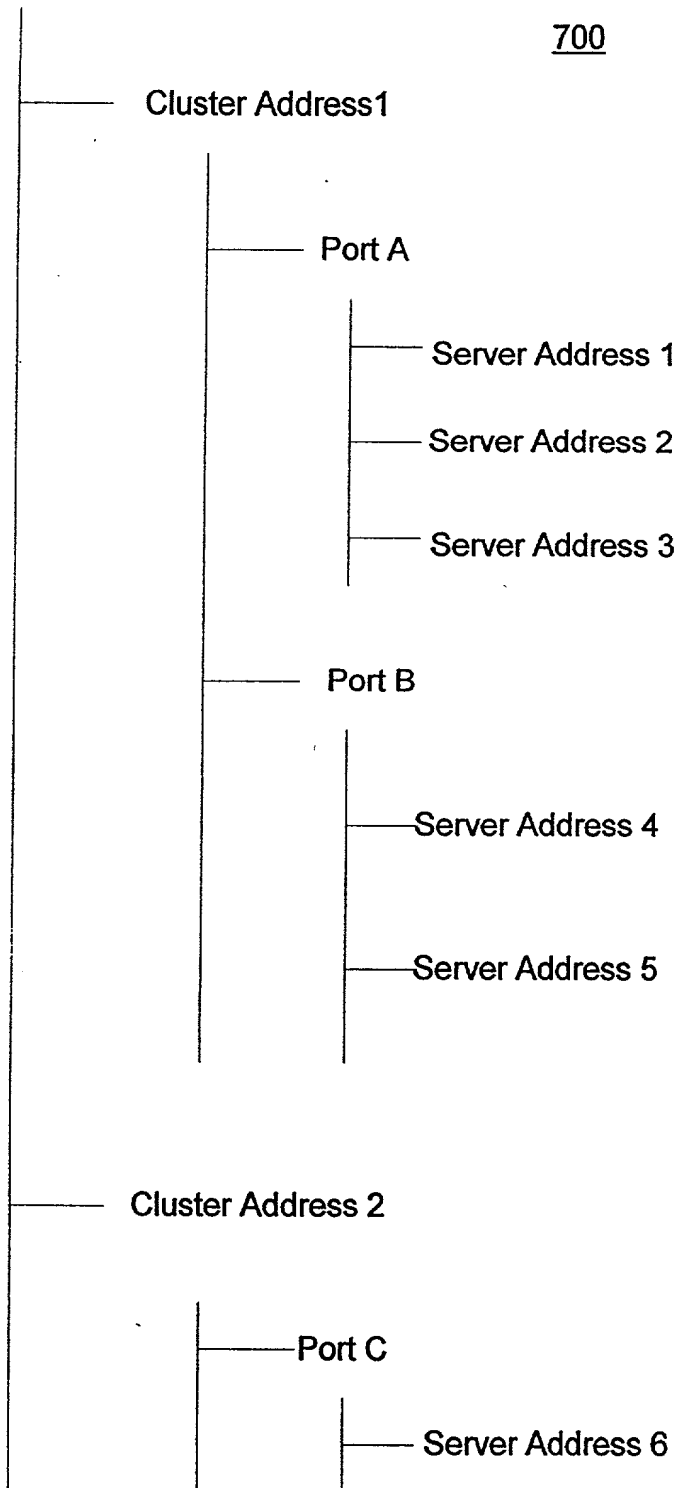
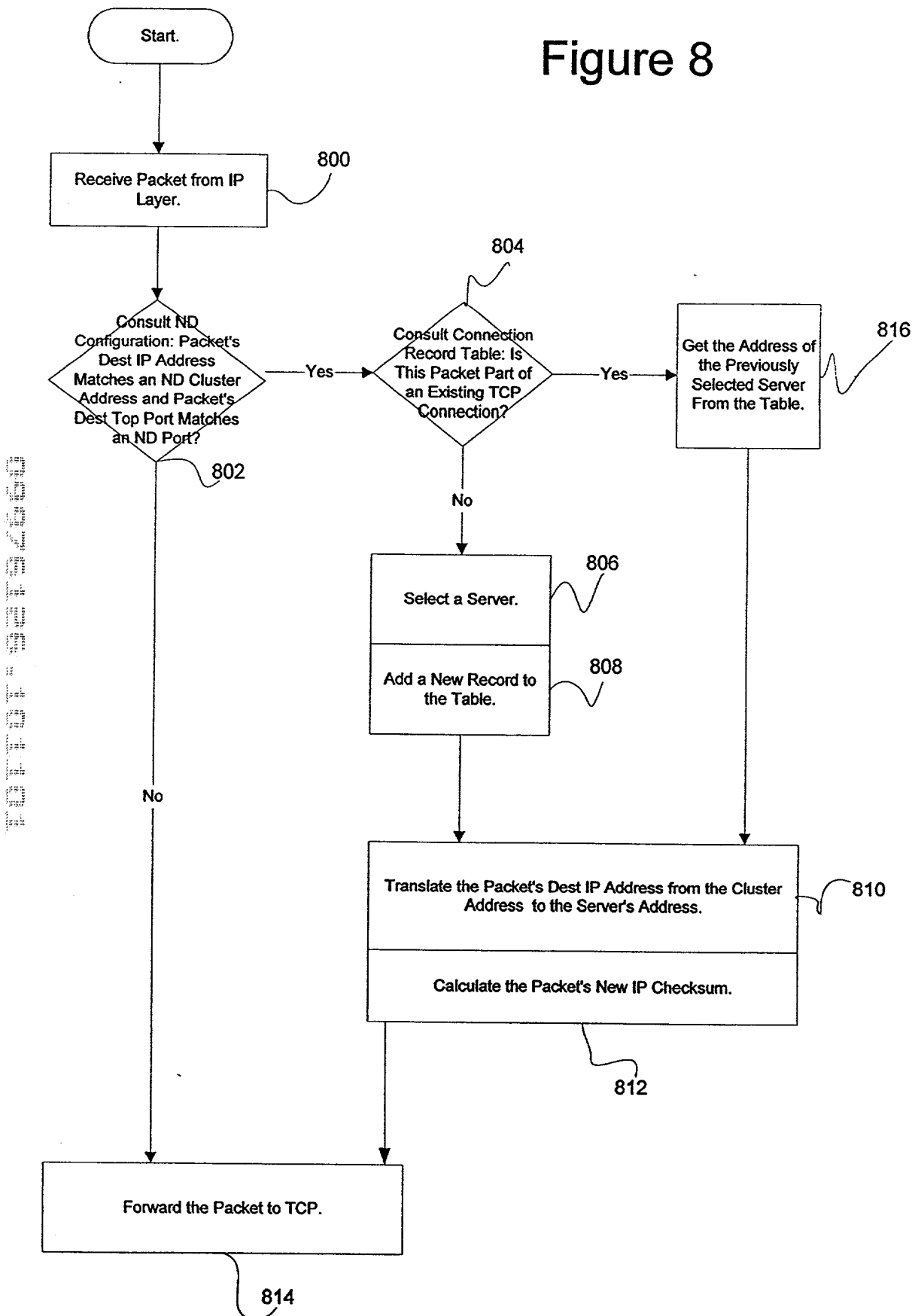


Figure 7

Figure 8



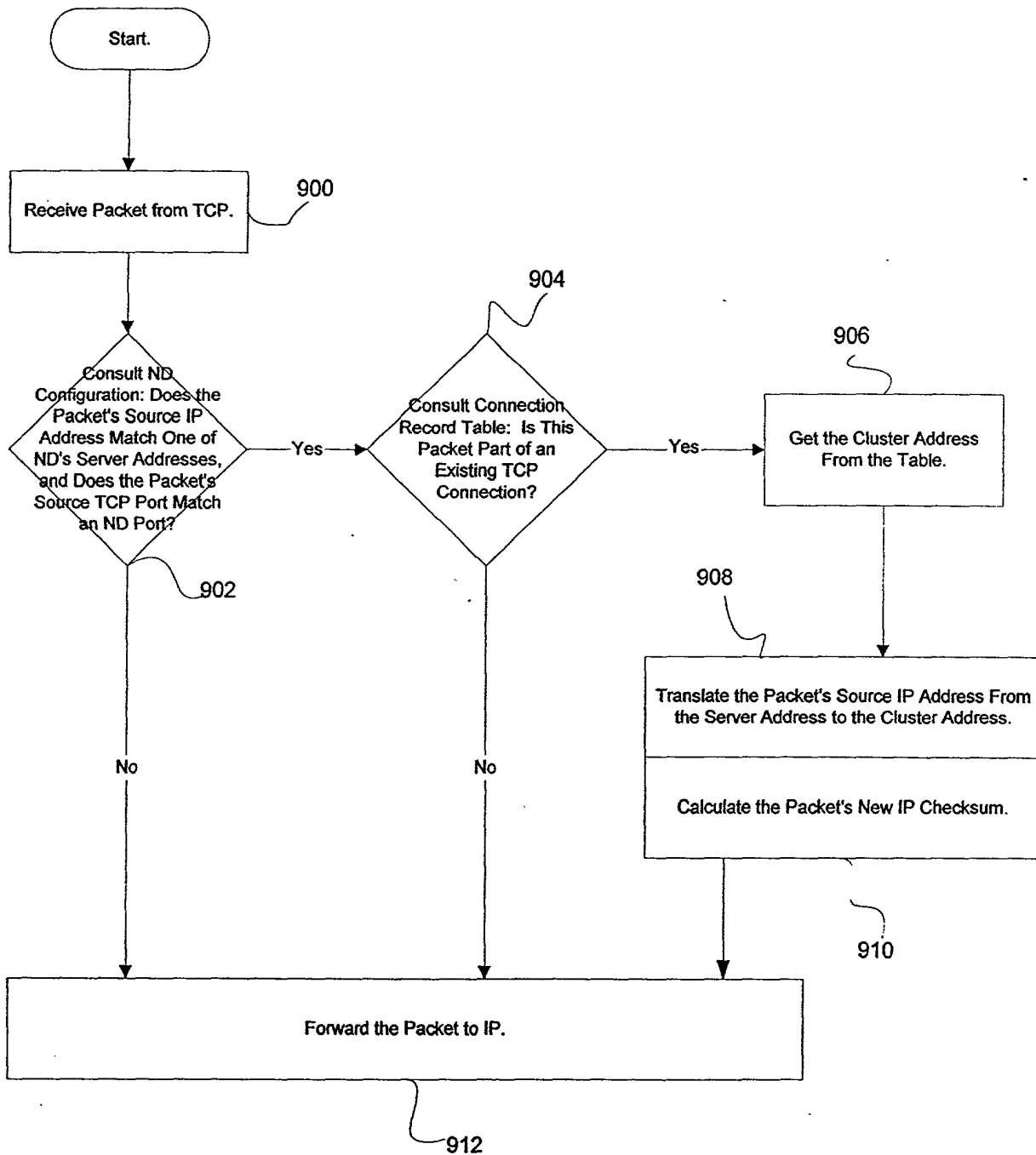


Figure 9

1000

1002	Source IP Address <u>1004</u>	Destination IP Address <u>1006</u>	Source TCP Port <u>1008</u>	Destination TCP Port <u>1010</u>	Selected Server Daemon IP Address <u>1012</u>
1002					
1002					
1002					
1002					

Figure 10

CR9-98-041

Declaration and Power of Attorney for Patent Application

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method and Apparatus Load Balancing Server Daemons Within a Server

the specification of which (check one)



is attached hereto.



was filed on _____ as Application Serial No. _____ and was amended on
_____.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Number

Country

Day/Month/Year

Priority Claimed

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications:

Serial No.

Filing Date

Status

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

A.B. Clay, Reg. No. 32,121; G. M. Doudnikoff, Reg. No. 32,847; E. H. Duffield, Reg. No. 25,970; J. W. Herndon, Reg. No. 27,901; J. S. Ray-Yarletts, Reg. No. 39,808;

Send all correspondence to: Gregory M. Doudnikoff
IBM Corporation, Dept. T81/062
3039 Cornwallis Road
RTP, NC 27709
919-254-1288
FAX: 254-4330

(1) Inventor: **Andrew Dwight Dingsor**
Signature: *Andrew Dwight Dingsor* 9-25-98
902 Huntsman Drive
Durham, North Carolina 27713
Residence:
Citizenship: USA
Post Office Address: same
(2) Inventor: **Stephen Manuel Fontes**
Signature: *Stephen Manuel Fontes* 9-25-98
1902 Loganwood Drive
Apex, North Carolina 27502
Residence:
Citizenship: USA
Post Office Address: same

PATENT
IBM Docket No. CR9-98-041

In the United States Patent and Trademark Office

Date: October 9, 2001

In re Application of: Dingsor, A. et al

Filed: Sept. 28, 1998

Serial Number: 09/161,905

For: Method and Apparatus Load Balancing Server Daemons within a Server

Art Unit: 2152

Examiner: Cardone, J.


Associate Power of Attorney

Please recognize, as additional attorneys, Duke W. Yee, Reg. No. 34,285; Colin P. Cahoon, Reg. No. 38,836; Stephen R. Loe, Reg. No. 43,757; Stephen J. Walder, Jr., Reg. No. 41,534; Stephen R. Tkacs, Reg. No. 46,430; Christopher P. O'Hagan, Reg. No. 46,966; Lisa L.B. Yociss, Reg. No. 36,975; Michael Nichols, Reg. No. 46,959; Patrick Holmes, Reg. No. 46,380; and Betty G. Formby, Reg. No. 36,536 in the above-identified patent application.

Please direct all communications on this patent application to:

Carstens, Yee & Cahoon, LLP
13760 Noel Road
Suite 900
Dallas, Texas 75240

Respectfully Submitted,


Gregory M. Doudnikoff, No. 32,847
ATTORNEY OF RECORD

IBM Corporation T81/503
P.O. Box 12195
Research Triangle Park, NC 27709
Off: (919) 254 - 1288